

BAB III

ANALISA, PERANCANGAN DAN IMPLEMENTASI SISTEM

3.1 Analisa Sistem

Analisa sistem merupakan kegiatan penguraian suatu sistem informasi yang utuh ke dalam bagian-bagian atau komponen-komponen komputer yang bertujuan untuk mengidentifikasi dan mengevaluasi masalah-masalah yang muncul, hambatan-hambatan yang mungkin terjadi dan kebutuhan-kebutuhan yang diharapkan sehingga mengarah kepada suatu solusi untuk perbaikan maupun pengembangan ke arah yang lebih baik dan sesuai dengan kebutuhan perkembangan teknologi. Analisa sistem dalam perancangan sistem informasi laba rugi pada usaha laundry menggunakan *flowchart*.

3.2 Pengguna Sistem

Pengguna sistem pada sistem informasi laba rugi pada usaha laundry adalah sebagai pemilik usaha laundry :

Untuk melakukan pencatatan dalam mencatat setiap pemasukan dan pengeluaran yang dilakukan untuk usaha laundry dan menghasilkan laporan laba rugi yang bisa di-*monitor* setiap saat.

3.3 Perancangan Sistem

Perancangan sistem dibuat bertujuan untuk memberikan gambaran secara umum kepada pengguna supaya mempermudah dalam perhitungan total biaya pemasukan, pengeluaran dan laba rugi dengan menggunakan *flowchart*.

3.3.1 Flowchart Perhitungan

Pada Flowchart ini menjelaskan tentang bagaimana perhitungan biaya pada transaksi Order Laundry dan transaksi Pengeluaran.

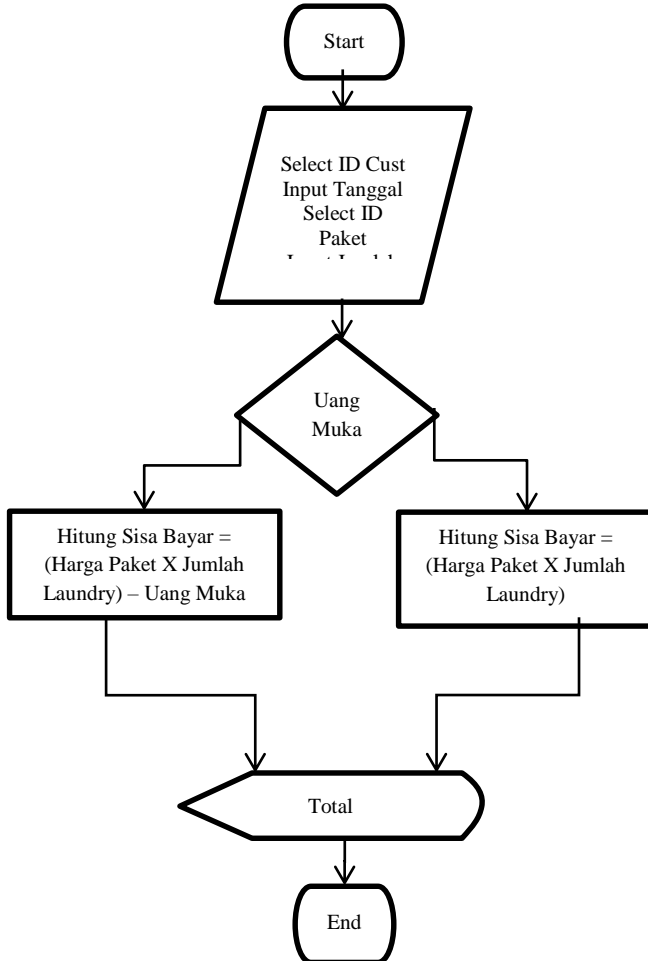
Keterangan Gambar 3.1 Flowchart Perhitungan :

Pada alur flowchart tersebut menjelaskan bagaimana cara menghitung biaya pada transaksi Order Laundry. Hal pertama yang dilakukan adalah menginput *ID customer*, *Tanggal Pesan* dan *Tanggal Selesai*. Kemudian ada *field* tersendiri untuk input *Jenis Paket* dan jumlah pakaian yang akan dilaundry-kan (dalam hal jumlah, kita asumsikan pakai satuan *kilogram*). Dari sini sudah terlihat secara otomatis berapa biaya yang harus dibayar customer. Namun customer tidak harus membayar seluruhnya didepan. Ada *field* tambahan untuk input uang muka dan akan langsung otomatis muncul sisa harga yang harus dibayar saat pengambilan laundry.

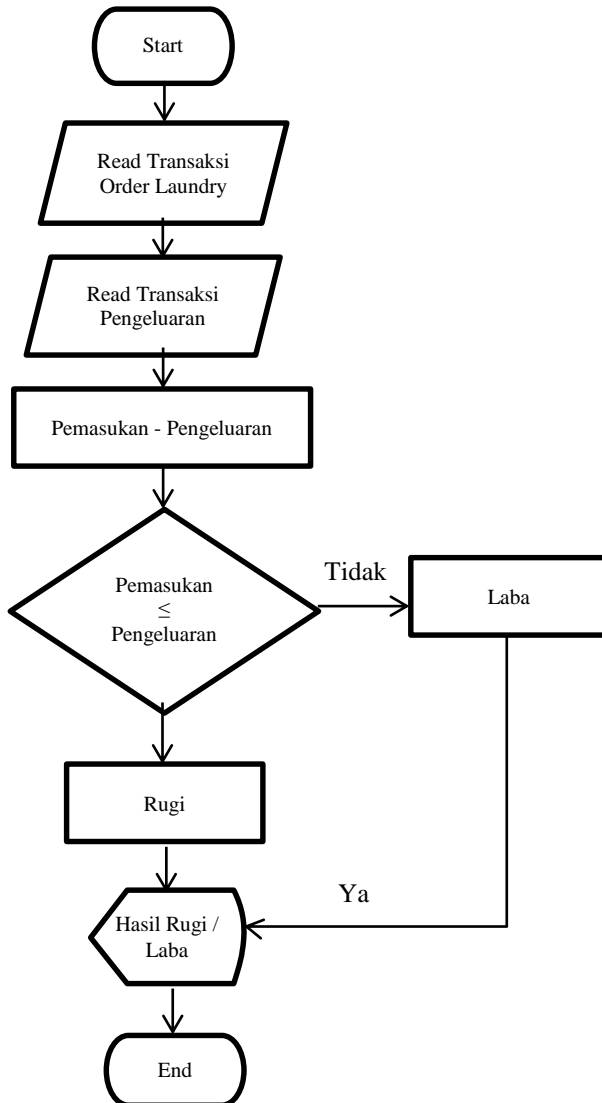
Keterangan Gambar 3.2 Flowchart Perhitungan :

Pada alur flowchart tersebut menjelaskan bagaimana cara menghitung atau menentukan laba rugi dalam suatu periode. Ini cukup sederhana karena perhitungannya pada dasarnya hanya menghitung jumlah dari transaksi pengeluaran dan transaksi order laundry. Pengeluaran yang dimaksud adalah untuk pembelian bahan seperti deterjen, pewangi, air, lalu untuk pembayaran air dan listrik, tentunya juga untuk pembayaran gaji karyawan. Dari semua transaksi pengeluaran yang ada akan dijumlahkan dengan transaksi order laundry, jika hasilnya *minus*

maka dalam periode tersebut mendapat kerugian dan jika hasilnya *plu* maka dalam periode tersebut mendapat laba.



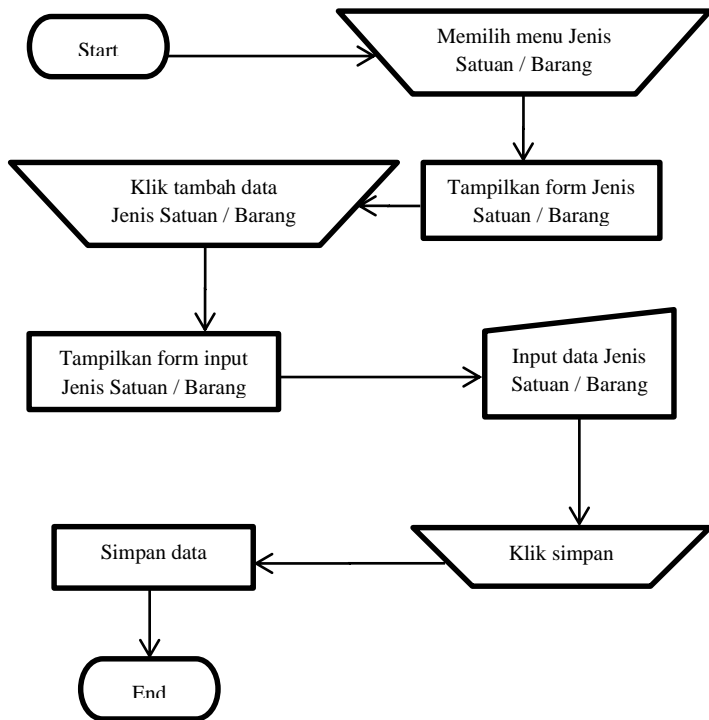
Gambar 3.1 Flowchart Perhitungan Uang Muka



Gambar 3.2 Flowchart Perhitungan Laba Rugi

3.3.2 Flowchart Input Data

Pada flowchart ini menjelaskan tentang input data satuan, pengguna, jenis cucian (paket), pelanggan (customer), barang, transaksi order laundry dan transaksi pengeluaran.



Gambar 3.3 Flowchart Input Data Jenis Satuan dan Barang

Keterangan Gambar 3.3 Flowchart Input Data Jenis Satuan dan barang :

Pada gambar 3.3 dapat dilihat alur dari proses Input Data Jenis Satuan dan Barang. Input Data Jenis Satuan dan Barang memiliki alur yang sama, oleh karena itu saya jelaskan dalam satu alur flowchart. Dimulai dari memilih menu Jenis Satuan / Barang. Kemudian form Jenis Satuan / Barang ditampilkan. Klik tambah data untuk input data baru. Akan muncul form input data. Ketikkan data di field yang sudah tersedia. Dan terakhir klik save.

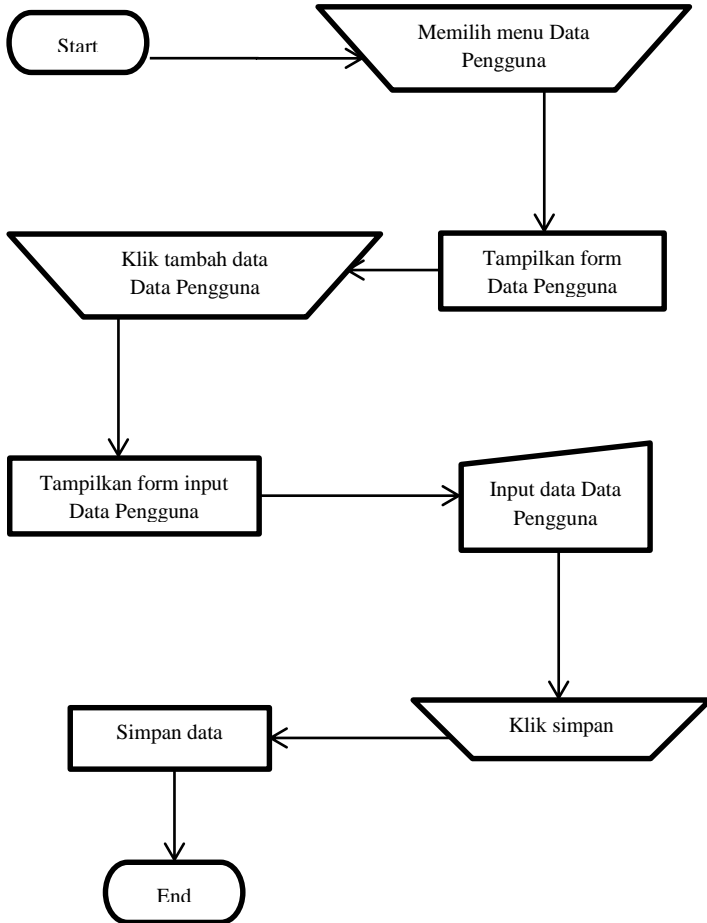
Keterangan Gambar 3.4 Flowchart Input Data Pengguna :

Pada gambar 3.4 dapat dilihat alur dari proses Input Data Pengguna. Dimulai dari memilih menu data pengguna, akan muncul form data pengguna. Klik tambah data untuk input data baru, akan muncul form input. Input data nama, username, password dan level penggunanya. Di data pengguna ini kita sediakan 2 level saja. Admin dan Kasir. Perbedaannya terletak pada menu Utility untuk backup dan restore data.

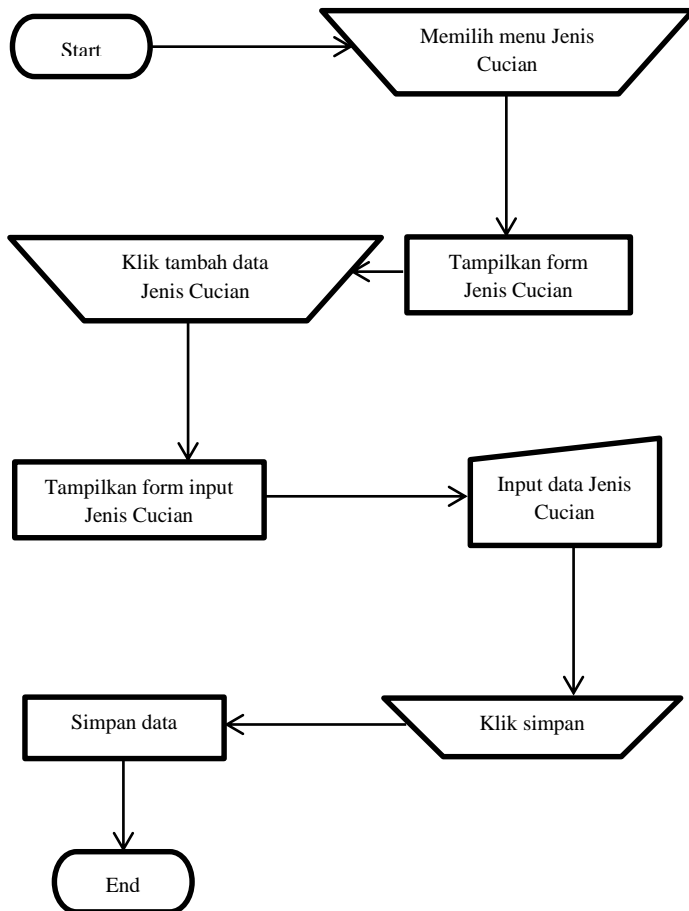
Keterangan Gambar 3.5 Flowchart Input Data Jenis Cucian (Paket) :

Pada gambar 3.5 dapat dilihat alur dari proses Input Data Jenis Cucian. Dimulai dari memilih menu Jenis Cucian, akan muncul form Jenis Cucian. Klik tambah data untuk input

data baru, akan muncul form input data. Input data Nama Jenis Cucian, Harga dan Satuannya. Klik save untuk menyimpan data.



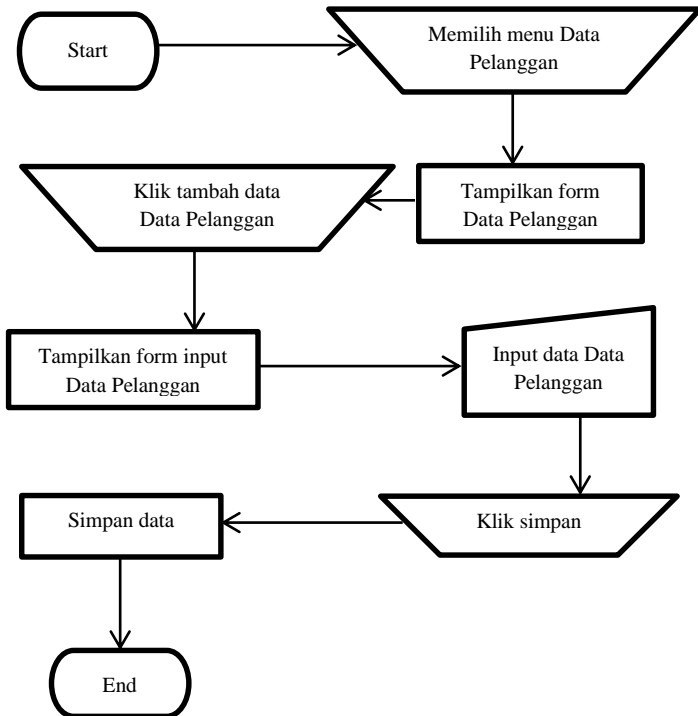
Gambar 3.4 Flowchart Input Data Pengguna



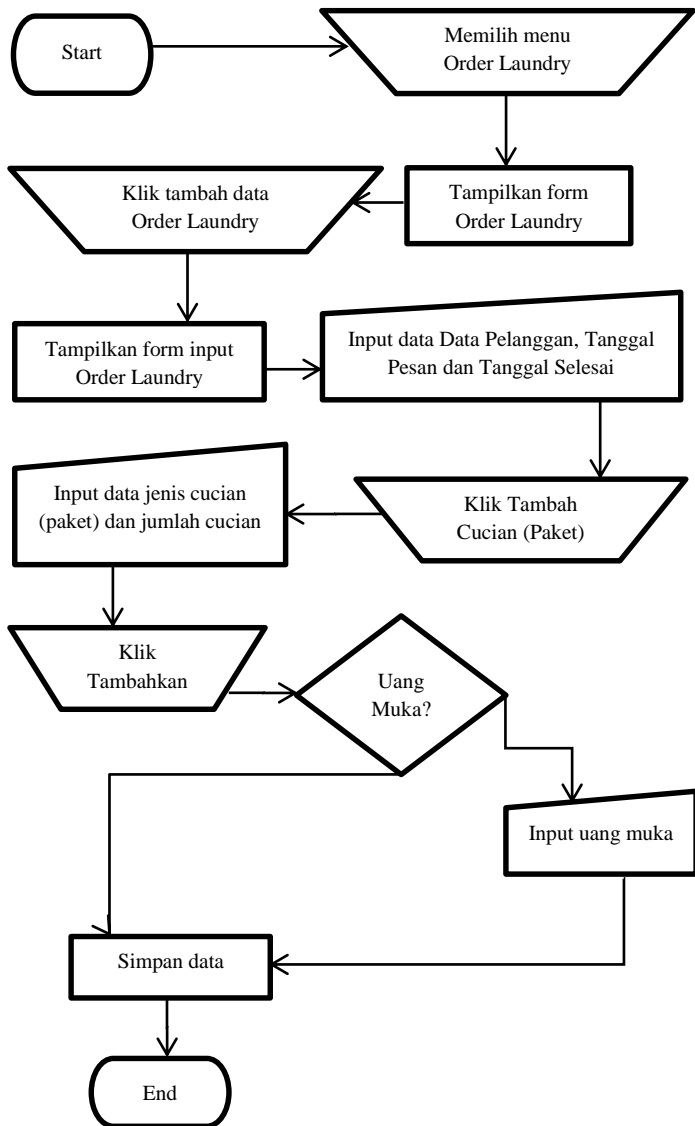
Gambar 3.5 Flowchart Input Data Jenis Cucian

Keterangan Gambar 3.6 Flowchart Input Data Pelanggan :

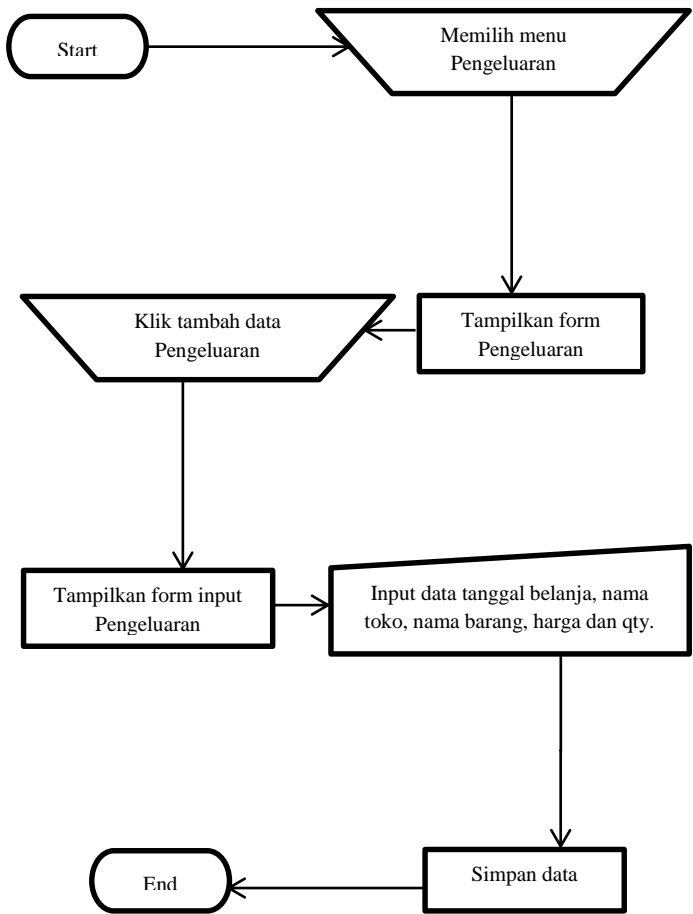
Pada gambar 3.6 dapat dilihat alur dari proses Input Data Pelanggan. Dimulai dari memilih menu Data Pelanggan, akan muncul form Data Pelanggan. Klik tambah data untuk input data baru, akan muncul form input data. Input data Nama Pelanggan, Nomer Telephone dan Alamatnya. Klik save untuk menyimpan data.



Gambar 3.6 Flowchart Input Data Pelanggan



Gambar 3.7 Flowchart Input Transaksi Order Laundry



Gambar 3.8 Flowchart Input Transaksi Pengeluaran

Keterangan Gambar 3.7 Flowchart Input Transaksi Order Laundry :

Pada gambar 3.7 dapat dilihat alur dari proses Input Transaksi Order Laundry. Dimulai dari memilih menu Transaksi Order Laundry, akan muncul form Order Laundry. Klik tambah data untuk input data baru, akan muncul form input data. Input data nama pelanggan, tanggal pesan dan tanggal selesai. Klik tambah cucian (paket), akan muncul form input paket cucian. Input jenis cucian (paket) dan jumlah kilo cucianya. Otomatis akan keluar harga total yang harus dibayar pelanggan. Klik tambahkan. Ada field untuk masukkan uang muka. Input uang muka jika ada dan akan otomatis keluar sisa harga yang harus dibayar saat pengambilan cucian. Jika tidak ada uang muka, langsung klik save untuk menyimpan.

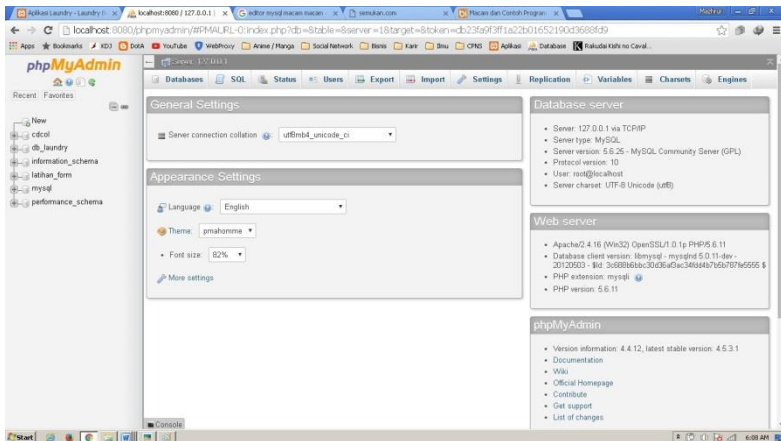
Keterangan Gambar 3.8 Flowchart Input Transaksi Pengeluaran :

Pada gambar 3.8 dapat dilihat alur dari proses Input Transaksi Pengeluaran. Dimulai dari memilih menu Transaksi Pengeluaran, akan muncul form Pengeluaran. Klik tambah data untuk input data baru, akan muncul form input data. Input data tanggal belanja, nama toko tempat belanja, barang yang dibeli, harga satuannya dan jumlah pembelian. Klik save untuk menyimpan. Maka akan muncul harga total dari pengeluaran tersebut.

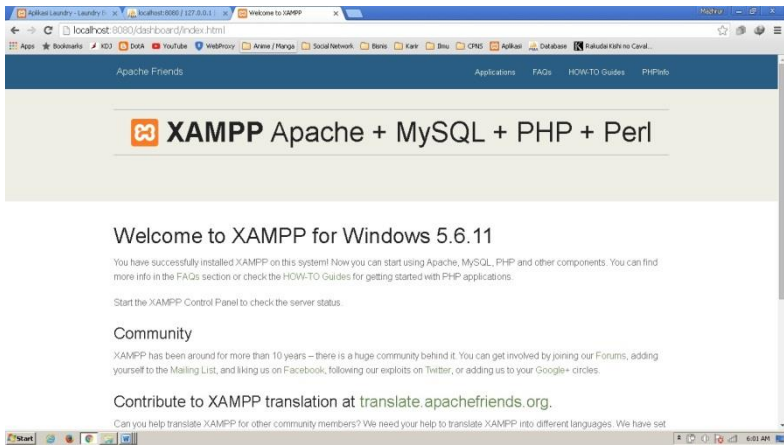
3.4 Perancangan Basis Data

Database atau basis data merupakan kumpulan dari data yang saling berhubungan satu dengan yang lainnya, tersimpan di dalam media penyimpanan dan digunakan perangkat lunak tertentu untuk memanipulasinya. *Database* merupakan suatu komponen yang penting dalam sistem bagi para pemakainya.

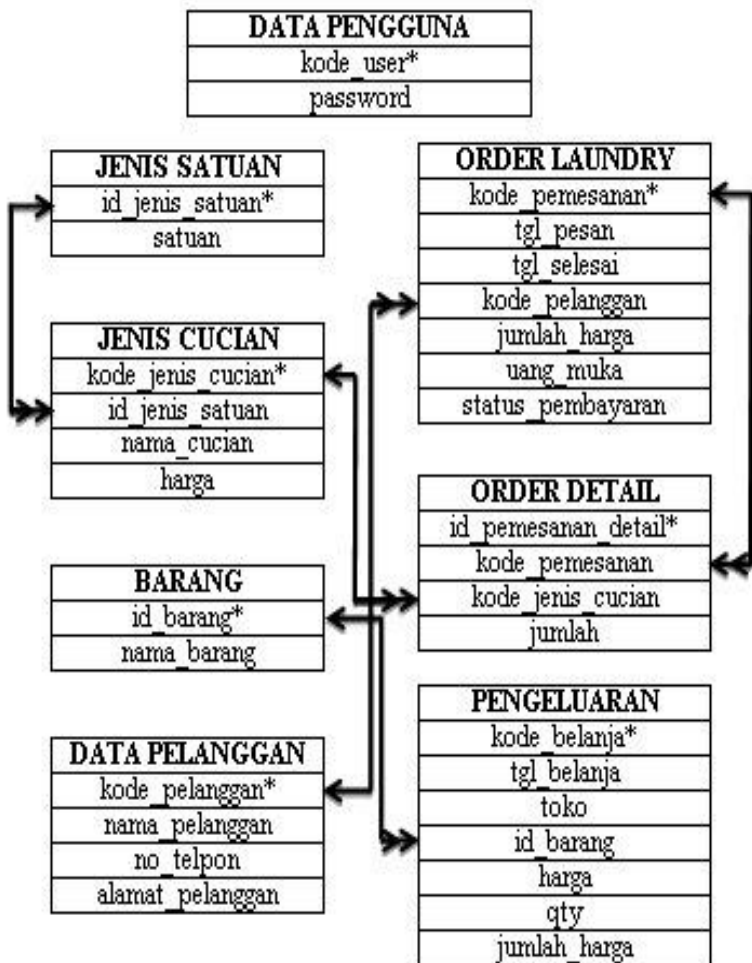
Pembuatan *database* dalam sistem ini menggunakan aplikasi MySQL versi 5.6, yang tergabung pada aplikasi XAMPP. Aplikasi MySQL itu sendiri adalah aplikasi open source yang berfungsi secara khusus menangani semua keperluan manusia yang berhubungan dengan *database*.



Gambar 3.9 Antar Muka MySQL Dengan Editor phpMyAdmin



Gambar 3.10 Antar Muka XAMPP



Gambar 3.11 Relasi Tabel Database

3.5 Perancangan Sistem Online

Sebuah aplikasi akan sangat efisien jika bisa digunakan secara online. Ini juga berfungsi apabila user membuka cabang baru yang berbeda lokasi. User tidak perlu repot – repot untuk pergi ke cabang baru, cukup dengan menggunakan internet dan otomatis bisa me-*monitor* usahanya tersebut.

Untuk membuat sebuah aplikasi bisa digunakan secara online kita membutuhkan bantuan software VPN gratis yaitu *Hamachi*.

3.5.1 Pengertian Hamachi

Hamachi adalah zero-configuration virtual private network (VPN) atau jaringan pribadi virtual tanpa konfigurasi. Hamachi merupakan aplikasi bagi-pakai yang didistribusikan secara bebas yang mampu membangun hubungan secara langsung antar-komputer yang terkoneksi ke dalam jaringan internet tanpa ada konfigurasi yang rumit, dengan kata lain, membentuk hubungan antar-komputer yang saling berjauhan melalui jaringan internet, seakan-akan komputer tersebut terhubung ke dalam jaringan lokal atau LAN.

3.5.2 Penggunaan Hamachi

Untuk bisa menggunakan *Hamachi* kita harus mendaftar terlebih dulu di website resminya. Untuk versi gratisnya, *Hamachi* bisa digunakan untuk 5 komputer user dan harus di-install di masing – masing komputer. Setelah mendaftar, kita harus membuat sebuah *network* untuk mendapatkan ID

3.6 Perancangan Framework CodeIgniter

Framework secara sederhana dapat diartikan kumpulan dari fungsi - fungsi / prosedur - prosedur dan *class - class* untuk tujuan tertentu yang sudah siap digunakan sehingga bisa lebih mempermudah dan mempercepat pekerjaan seorang programmer, tanpa harus membuat fungsi atau class dari awal.

Ada beberapa alasan mengapa menggunakan Framework :

- Mempercepat dan mempermudah pembangunan sebuah aplikasi web.
- Relatif memudahkan dalam proses maintenance karena sudah ada pola tertentu dalam sebuah framework (dengan syarat programmer mengikuti pola standart yang ada)
- Umumnya framework menyediakan fasilitas - fasilitas yang umum dipakai sehingga kita tidak perlu membangun dari awal (misalnya validasi, ORM, pagination, multiple database, scaffolding, pengaturan session, error handling, dll)
- Lebih bebas dalam pengembangan jika dibandingkan CMS

Sejauh ini, framework untuk pemrograman website yang sering digunakan adalah framework php dan framework css. Contoh framework php adalah CodeIgniter dan Zend Framework. Sedangkan, contoh framework css adalah Bootstrap.

3.6.1 Pengertian Framework CodeIgniter

CodeIgniter (CI) sebuah framework untuk programmer yang ingin membangun website dengan PHP. CI merupakan langkah cepat untuk membuat sebuah website daripada membangun sebuah website dengan alur koding yang biasa.

Dengan mempacketkan library-library, alur pemrograman akan terstruktur dan logika bisnis/proses akan lebih mudah dimengerti oleh developer/programmer lain. CI telah mempacketkan itu semua sehingga mampu meminimalisirkan jumlah koding yang dibangun.

Kelebihan dari CodeIgniter :

- Gratis.
CodeIgniter dilisensikan dibawah lisensi Apache/BSD style open source license, ini berarti kita dapat menggunakannya sesuai dengan keinginan kita.
- Berjalan di PHP versi 4 dan 5.
Sekarang ini PHP sudah mencapai versi ke 5, meskipun begitu masih banyak orang yang tetap menggunakan PHP versi 4, oleh sebab itu CodeIgniter dikembangkan agar tetap kompatibel dengan PHP versi 4 dan dapat dijalankan pada PHP versi 5.
- Ringan dan cepat.
Secara default CodeIgniter hanya berjalan dengan me load beberapa pustaka saja, dengan demikian hanya membutuhkan resource yang sedikit sehingga ringan dan cepat dijalankan. Pustaka-pustaka lain yang nantinya akan digunakan bisa di load sesuai dengan kebutuhan.
- Menggunakan MVC.
CodeIgniter menggunakan lingkungan pengembangan dengan metode Model View Controller (MVC) yang membedakan antara logika dan presentasi/tampilan, sehingga

tugas bisa lebih mudah dipecah-pecah. Ada bagian yang khusus membuat tampilan dan bagian yang membuat core programnya.

- Dokumentasi.

Salah satu hal yang bisa dijadikan barometer apakah sebuah aplikasi benar-benar dikembangkan atau tidak bisa dilihat dari dokumentasinya. Dalam hal ini CodeIgniter sangat luar biasa, terdapat dokumentasi yang sangat lengkap tentang semua hal yang ada dalam CodeIgniter. Mulai dari langkah instalasi sampai dokumentasi fungsi-fungsinya tersedia. Adanya dokumentasi sangat memudahkan bagi pemula dalam mempelajari lingkungan pengembangan website dengan CodeIgniter.

- Pustaka yang lengkap.

CodeIgniter dilengkapi dengan berbagai pustaka siap pakai untuk berbagai kebutuhan, misalnya saja koneksi database, email, session dan cookies, keamanan, manipulasi gambar dan banyak lagi.

3.6.2 Pemasangan Framework CodeIgniter

Untuk pemasangan awal framework, tentunya kita harus download dahulu master codeigniter-nya dari internet. Biasanya hasil download berupa file berekstensi .zip. Lalu ekstrak file tersebut ke dalam folder htdoc. Rubah nama folder hasil ekstrak menjadi codeigniter saja. Untuk mengetes apakah codeigniter sudah dapat dijalankan, anda dapat mengetik URL

pada browser : <http://localhost/codeigniter/> maka akan muncul tulisan seperti berikut :



Gambar 3.14 Tampilan Awal Codeigniter

3.6.3 Konfigurasi Database CodeIgniter

Agar aplikasi dapat berjalan lancar, maka kita harus mengatur *konfigurasi* database menyesuaikan dengan settingan dari web server. Silahkan buka file database.php yang ada pada `application/config/database.php`.

```

73 $active_group = 'default';
74 $query_builder = TRUE;
75
76 $db['default'] = array(
77     'dsn' => '',
78     'hostname' => 'localhost',
79     'username' => '',
80     'password' => '',
81     'database' => '',
82     'dbdriver' => 'mysqli',
83     'dbprefix' => '',
84     'pconnect' => FALSE,
85     'db_debug' => (ENVIRONMENT !== 'production'),
86     'cache_on' => FALSE,
87     'cachedir' => '',
88     'char_set' => 'utf8',
89     'dbcollat' => 'utf8_general_ci',
90     'swap_pre' => '',
91     'encrypt' => FALSE,
92     'compress' => FALSE,
93     'stricton' => FALSE,
94     'failover' => array(),
95     'save_queries' => TRUE
96 );
97

```

Gambar 3.15 Konfigurasi Database

Keterangan :

- 'hostname' = diisi dengan localhost karena kita bekerja dengan xampp
- 'username' = sesuai username phpmyadmin
- 'password' = sesuai password phpmyadmin
- 'database' = sesuai nama database yang sudah dibuat

3.6.4 Konfigurasi Autoload CodeIgniter

Untuk mengelola database di codeigniter, kita juga harus atur autoload-nya di application/config/autoload.php menjadi seperti dibawah ini :

```
60  */
61  $autoload['libraries'] = array('database');
62  $autoload['helper'] = array('form');
63  |
```

Gambar 3.16 Konfigurasi Autoload

Keterangan :

‘libraries’ = untuk me-load database

‘helper’ = untuk me-load form

3.6.5 Konfigurasi Routes CodeIgniter

File routes.php juga terletak di application/config/routes.php. Pada file ini yang harus disetting hanyalah script :

```
54
55  $route['default_controller'] = 'welcome';
56
57
```

Gambar 3.17 Konfigurasi Routes

Keterangan :

Standartnya, variable tersebut akan terisi dengan “welcome” yang akan menampilkan halaman utama CodeIgniter. Variable tersebut harus diisi dengan controller utama yang akan digunakan sebagai tampilan utama website.

3.6.6 Konfigurasi Config CodeIgniter

File config .php yang dimaksud adalah file yang terletak pada lokasi application/config/config.php, pada file ini yang harus diubah hanya beberapa variable saja seperti berikut :

```
27  
28 $config['base_url'] = '';  
29 $config['index_page'] = 'index.php';  
30
```

Gambar 3.18 Konfigurasi Config

Keterangan :

\$config['base_url'] harus diisi dengan alamat path dari instalasi codeIgniter, sehingga pada saat aplikasi diakses, maka akan diarahkan ke alamat tersebut. \$config['index_page'] dapat dikosongkan atau diisi

dengan index.php, tapi sebaiknya dikosongkan saja karena akan menghapus index.php agar lebih SEO.

3.7 Implementasi Database

Implementasi database dilakukan berdasarkan hasil dari tahap perancangan sistem. Dan sebagai tempat penyimpanan data untuk setiap inputan seperti tabel data pengguna, tabel jenis cucian (paket), tabel data pelanggan, tabel barang, tabel order laundry dan tabel pengeluaran.

Berikut adalah *query* yang digunakan untuk membuat database sistem informasi laba rugi pada usaha laundry.

CREATE DATABASE Laundry :

Adapun tabel yang digunakan di dalam database sistem informasi, tabel – tabel tersebut diantaranya :

3.7.1 Tabel Data Pengguna

```
CREATE TABLE `user` (  
  `kode_user` int(5) NOT NULL AUTO_INCREMENT,  
  `username` varchar(100) NOT NULL,  
  `password` varchar(100) NOT NULL,  
  `level` varchar(10) NOT NULL,  
  `nama_user` varchar(100) NOT NULL,  
  PRIMARY KEY (`kode_user`)  
) ENGINE=MyISAM AUTO_INCREMENT=4 DEFAULT CHARSET  
=latin1
```

Gambar 3.19 Query Tabel Data Pengguna

Gambar 3.19 merupakan *query* untuk membuat tabel data pengguna dan atribut yang dibutuhkan untuk penyimpanan data. Dan hasil *generate query* di atas adalah sebagai berikut.

Tabel 3.1 : Data Pengguna

Column	Type	Collation	Null	Default	Extra
kode_user	int(5)		No	<i>None</i>	AUTO_INCREMENT
username	varchar(100)	latin1_swedish_ci	No	<i>None</i>	
password	varchar(100)	latin1_swedish_ci	No	<i>None</i>	
level	varchar(10)	latin1_swedish_ci	No	<i>None</i>	
nama_user	varchar(100)	latin1_swedish_ci	No	<i>None</i>	

3.7.2 Tabel Jenis Cucian (Paket)

```
CREATE TABLE `jenis_cucian` (
  `kode_jenis_cucian` int(5) NOT NULL AUTO_INCREMENT,
  `id_jenis_satuan` int(10) NOT NULL,
  `nama_cucian` varchar(100) NOT NULL,
  `harga` int(20) NOT NULL,
  PRIMARY KEY (`kode_jenis_cucian`)
) ENGINE=MyISAM AUTO_INCREMENT=6 DEFAULT CHARSET=latin1
```

Gambar 3.20 Query Tabel Jenis Cucian (Paket)

Gambar 3.20 merupakan *query* untuk membuat tabel jenis cucian (paket) dan atribut yang dibutuhkan untuk penyimpanan data. Dan hasil *generate query* di atas adalah sebagai berikut.

Tabel 3.2 : Jenis Cucian (Paket)

Column	Type	Collation	Null	Default	Extra
kode_jenis_cucian	int(5)		No	None	AUTO_INCREMENT
id_jenis_satuan	int(10)		No	None	
nama_cucian	varchar(100)	latin1_swedish_ci	No	None	
harga	int(20)		No	None	

3.7.3 Tabel Data Pelanggan

```
CREATE TABLE `pelanggan` (
  `kode_pelanggan` int(5) NOT NULL AUTO_INCREMENT,
  `nama_pelanggan` varchar(100) NOT NULL,
  `jenis` varchar(20) NOT NULL,
  `alamat_pelanggan` text NOT NULL,
  PRIMARY KEY (`kode_pelanggan`)
) ENGINE=MyISAM AUTO_INCREMENT=41 DEFAULT CHARSET=
T=latin1
```

Gambar 3.21 Query Tabel Data Pelanggan

Gambar 3.21 merupakan *query* untuk membuat tabel data pelanggan dan atribut yang dibutuhkan untuk penyimpanan data. Dan hasil *generate query* di atas adalah sebagai berikut.

Tabel 3.3 : Data Pelanggan

Column	Type	Collation	Null	Default	Extra
kode_pelanggan	int(5)		No	None	AUTO_INCREMENT
nama_pelanggan	varchar(100)	latin1_swedish_ci	No	None	
jenis	varchar(20)	latin1_swedish_ci	No	None	
alamat_pelanggan	text	latin1_swedish_ci	No	None	

3.7.4 Tabel Barang

```
CREATE TABLE `barang` (  
  `id_barang` int(11) NOT NULL AUTO_INCREMENT,  
  `nama_barang` varchar(50) NOT NULL,  
  PRIMARY KEY (`id_barang`)  
) ENGINE=InnoDB AUTO_INCREMENT=9 DEFAULT CHARSET=latin1
```

Gambar 3.22 Query Tabel Barang

Gambar 3.22 merupakan *query* untuk membuat tabel barang dan atribut yang dibutuhkan untuk penyimpanan data. Dan hasil *generate query* di atas adalah sebagai berikut.

Tabel 3.4 : Barang

Column	Type	Collation	Null	Default	Extra
id_barang	int(11)		No	None	AUTO_INCREMENT
nama_barang	varchar(50)	latin1_swedish_ci	No	None	

3.7.5 Tabel Order Laundry

```
CREATE TABLE `pemesanan` (  
  `kode_pemesanan` varchar(20) NOT NULL,  
  `tgl_pesan` varchar(30) NOT NULL,  
  `tgl_selesai` varchar(30) NOT NULL,  
  `kode_pelanggan` int(5) NOT NULL,  
  `jumlah_harga` int(20) NOT NULL,  
  `uang_muka` int(20) NOT NULL,  
  `status_pembayaran` varchar(50) NOT NULL DEFAULT 'Belum Lun  
as',  
  PRIMARY KEY (`kode_pemesanan`)  
) ENGINE=MyISAM DEFAULT CHARSET=latin1
```

Gambar 3.23 Query Tabel Pemesanan (Order Laundry)

```
CREATE TABLE `pemesanan_detail` (  
  `id_pemesanan_detail` int(10) NOT NULL AUTO_INCREMENT,  
  `kode_pemesanan` varchar(30) NOT NULL,  
  `kode_jenis_cucian` int(10) NOT NULL,  
  `jumlah` int(10) NOT NULL,  
  PRIMARY KEY (`id_pemesanan_detail`)  
) ENGINE=InnoDB AUTO_INCREMENT=2 DEFAULT CHARSET=l  
atin1
```

Gambar 3.24 Query Tabel Pemesanan Detail (Order Laundry)

Gambar 3.23 dan 3.24 merupakan *query* untuk membuat tabel order laundry dan atribut yang dibutuhkan untuk penyimpanan data. Dan hasil *generate query* di atas adalah sebagai berikut.

Tabel 3.5 : Pemesanan (Order Laundry)

Column	Type	Collation	Null	Default	Extra
kode_pemesanan	varchar(20)	latin1_swedish_ci	No	<i>None</i>	
tgl_pesan	varchar(30)	latin1_swedish_ci	No	<i>None</i>	
tgl_selesai	varchar(30)	latin1_swedish_ci	No	<i>None</i>	
kode_pelanggan	int(5)		No	<i>None</i>	
jumlah_harga	int(20)		No	<i>None</i>	
uang_muka	int(20)		No	<i>None</i>	
status_pembayaran	varchar(50)	latin1_swedish_ci	No	<i>Belum Lunas</i>	

Tabel 3.6 : Pemesanan Detail (Order Laundry)

Column	Type	Collation	Null	Default	Extra
id_pemesanan_detail	int(10)		No	None	AUTO_INCREMENT
kode_pemesanan	varchar(30)	latin1_swedish_ci	No	None	
kode_jenis_uci	int(10)		No	None	
jumlah	int(10)		No	None	

3.7.6 Tabel Pengeluaran

```
CREATE TABLE `pengeluaran` (
  `kode_belanja` int(11) NOT NULL AUTO_INCREMENT,
  `tgl_belanja` varchar(30) NOT NULL,
  `toko` varchar(50) NOT NULL,
  `id_barang` varchar(25) NOT NULL,
  `harga` varchar(20) NOT NULL,
  `qty` int(11) NOT NULL,
  `jumlah_harga` int(20) NOT NULL,
  PRIMARY KEY (`kode_belanja`)
) ENGINE=MyISAM AUTO_INCREMENT=15 DEFAULT CHARSET=latin1
```

Gambar 3.25 Query Tabel Pengeluaran

Gambar 3.25 merupakan *query* untuk membuat tabel pengeluaran dan atribut yang dibutuhkan untuk penyimpanan data. Dan hasil *generate query* di atas adalah sebagai berikut.

Tabel 3.7 : Pengeluaran

Column	Type	Collation	Null	Default	Extra
kode_belanja	int(11)		No	<i>None</i>	AUTO_INCREMENT
tgl_belanja	varchar(30)	latin1_swedish_ci	No	<i>None</i>	
toko	varchar(50)	latin1_swedish_ci	No	<i>None</i>	
id_barang	varchar(25)	latin1_swedish_ci	No	<i>None</i>	
harga	varchar(20)	latin1_swedish_ci	No	<i>None</i>	
qty	int(11)		No	<i>None</i>	
jumlah_harga	int(20)		No	<i>Belum Lunas</i>	

3.8 Implementasi Program

3.8.1 Source Code Laba Rugi

Gambar berikut merupakan *query* untuk *Source Code* Laba Rugi ke database aplikasi laundry.

```
<?php if ( ! defined('BASEPATH')) exit('No direct script access
allowed');
class laporan_laba extends CI_Controller {
    function index()
    {
        if($this->session->userdata("logged_in")!= "")
        {
            redirect("dashboard");
        }
        else
        {
            redirect("login");
        }
    }
}
```

```

        public function __construct() {
            parent::__construct();
            $this->load->database();
            $this->load->library('Excel_generator');
        }

        function harian($uri=0)
        {
            if($this->session->userdata("logged_in")!= "")
            {
                $scari =
                $this->session->userdata("tgl_cari");
                $d['dt_retrieve'] =
                $this->app_load_data_model->indexs_laporan_laba($scari,$GLOBALS['
                site_limit_medium'],$uri);
                $d['tgl_cari'] =
                $this->session->userdata("tgl_cari");

                $this->load->view($GLOBALS['site_theme']."/bg_header");

                $this->load->view($GLOBALS['site_theme']."/bg_left",$d);

                $this->load->view($GLOBALS['site_theme']."/laporan_laba/b
                g_home_harian");

                $this->load->view($GLOBALS['site_theme']."/bg_footer");

                $this->session->unset_userdata(array('tgl_cari'=>""));

                /*$this->session->unset_userdata('tgl_pesani');

                $this->session->unset_userdata('tgl_selesai');

                $this->session->unset_userdata('jumlah_harga');
                $this->cart->destroy();*/
            }
            else
            {
                redirect("login");
            }
        }
    }

```

```

function bulanan($Suri=0)
{
    if($this->session->userdata("logged_in")!= "")
    {
        $cari =
$this->session->userdata("bulan_cari");
        $d['dt_retrieve'] =
$this->app_load_data_model->indexs_laporan_laba($cari,$GLOBALS['
site_limit_medium'],$Suri);
        $d['bulan_cari'] =
$this->session->userdata("bulan_cari");

        $this->load->view($GLOBALS['site_theme']."/bg_header");

        $this->load->view($GLOBALS['site_theme']."/bg_left",$d);

        $this->load->view($GLOBALS['site_theme']."/laporan_laba/b
g_home_bulanan");

        $this->load->view($GLOBALS['site_theme']."/bg_footer");

        $this->session->unset_userdata(array('bulan_cari'=>));
    }
    else
    {
        redirect("login");
    }
}

function set()
{
    if($this->session->userdata("logged_in")!= "")
    {
        $st = $_POST['jenis'];
        if($st=="harian")
        {

            if(isset($_POST['tampil'])){

                $set['tgl_cari'] = $_POST['tgl'];
            }
        }
    }
}

```

```

        redirect("dashboard/laporan_laba/harian");
    }else
if(isset($_POST['print'])){
        $d =
$this->db->query("select * from pemesanan where tgl_pesan =
".$_POST['tgl']."");

        $this->excel_generator->set_query($d);

        $this->excel_generator->set_header(array('Kode Order',
'Tanggal masuk', 'Tanggal Selesai', 'Tarif Total','Uang Muka','Status'));

        $this->excel_generator->set_column(array('kode_pemesanan',
'tgl_pesan', 'tgl_selesai',
'jumlah_harga','uang_muka','status_pembayaran'));

        $this->excel_generator->set_width(array(3, 10, 10, 10, 10,
10));

        $this->excel_generator->exportTo2007('Laporan Order
harian');
    }
    }
    else if($st=="bulanan")
    {
if(isset($_POST['tampil'])){
        $set['bulan_cari'] = $_POST['bulan'];

        $this->session->set_userdata($set);

        redirect("dashboard/laporan_laba/bulanan");
    }else
if(isset($_POST['print'])){
        $d =
$this->db->query("select * from pemesanan where tgl_pesan like
%".$_POST['bulan']."%");

        $this->excel_generator->set_query($d);

```

```
        $this->excel_generator->set_column(array('kode_pemesanan',  
'tgl_pesan', 'tgl_selesai',  
'jumlah_harga','uang_muka','status_pembayaran'));  
  
        $this->excel_generator->set_width(array(3, 10, 10, 10, 10,  
10));  
  
        $this->excel_generator->exportTo2007('Laporan Order  
bulanan');  
    }  
    }  
    }  
    else  
    {  
        redirect("login");  
    }  
}
```

Gambar 3.26 Source Code Laba Rugi